

Boolean Operators

Important For Conditional Code: Return **true** or **false**

==	Check for value equality	e.g. a == b
!=	Check for value inequality	e.g. a != b
=== and !==	Check for value AND type (in)equality	e.g. a === b / a !== b
> & <	Check for value being greater / smaller	e.g. a > b / a < b
>= & <=	Check for value being greater or equal / smaller or equal	e.g. a >= b / a <= b
!	Check if NOT true	e.g. !a

Prefer over ==

Beware of Objects & Arrays in Comparisons!

{ name: 'Max' }

=== or ==

{ name: 'Max' }

false!

Objects and arrays are kind of special in JavaScript!



“Boolean Tricks” with Logical Operators

Boolean Coercion via double NOT (double bang) operator

!!

e.g. !!`""`, e.g. !!1

false, true

Default value assignment via OR operator

||

e.g. `const name = somelInput || 'Max'`

somelInput if not falsy, 'Max' otherwise

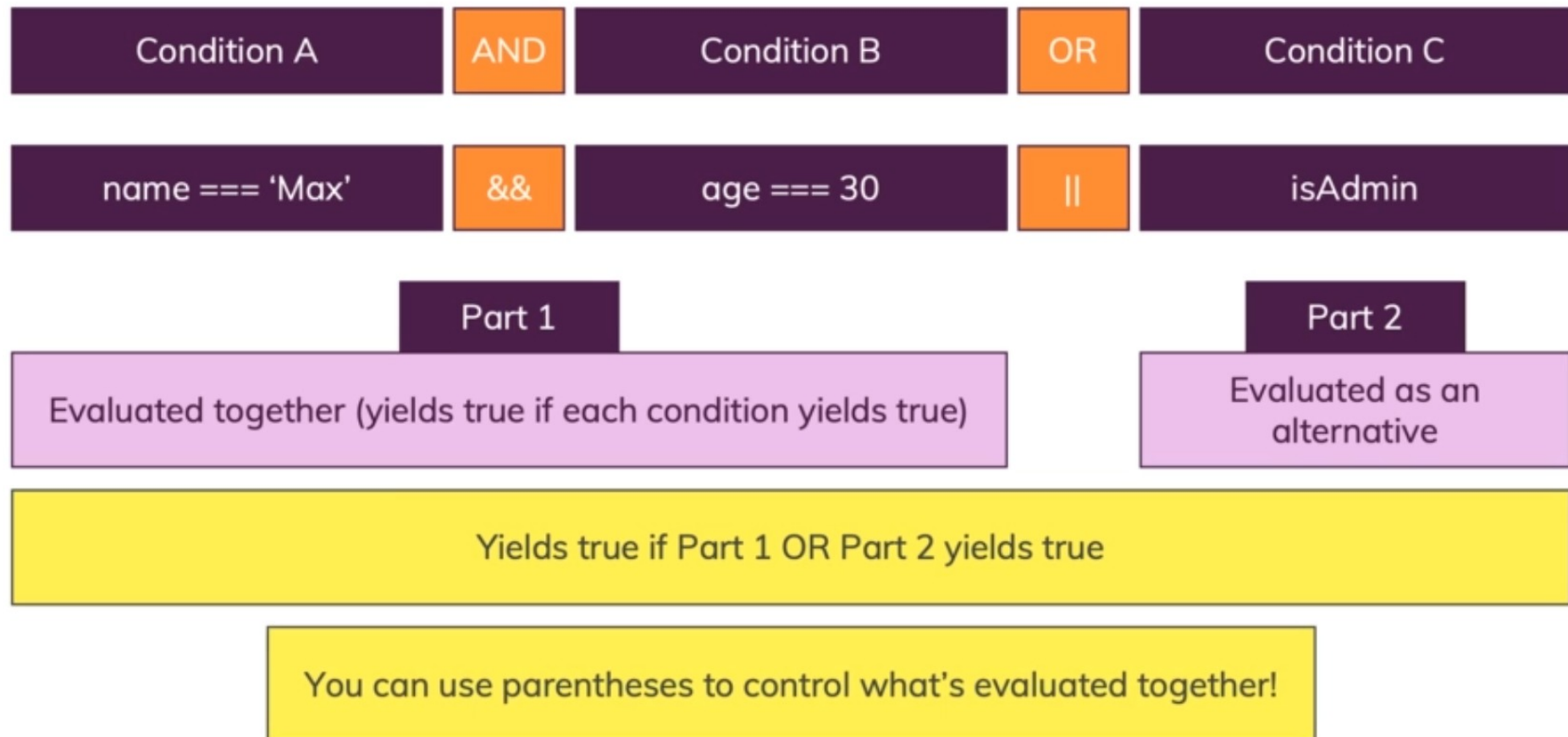
Use value if condition is true via AND operator

&&

e.g. `const name = isLoggedIn && 'Max'`

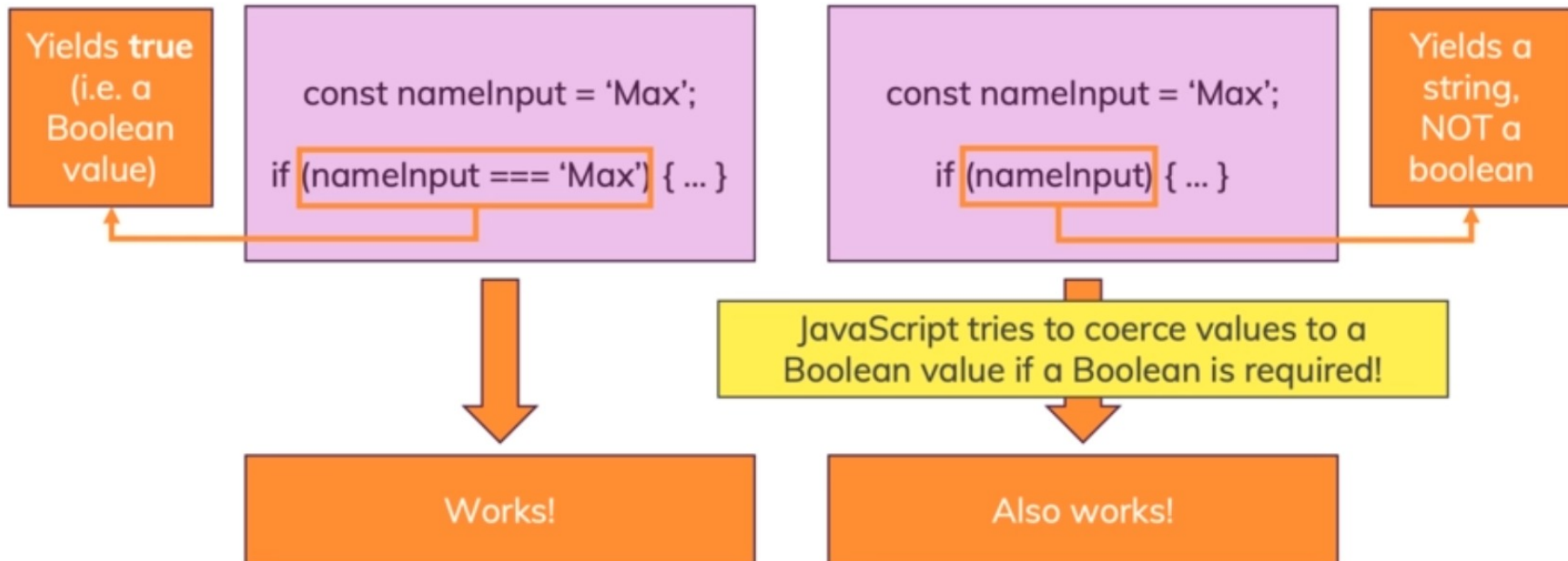
'Max' is set if isLoggedIn is true, false otherwise

Combining Conditions

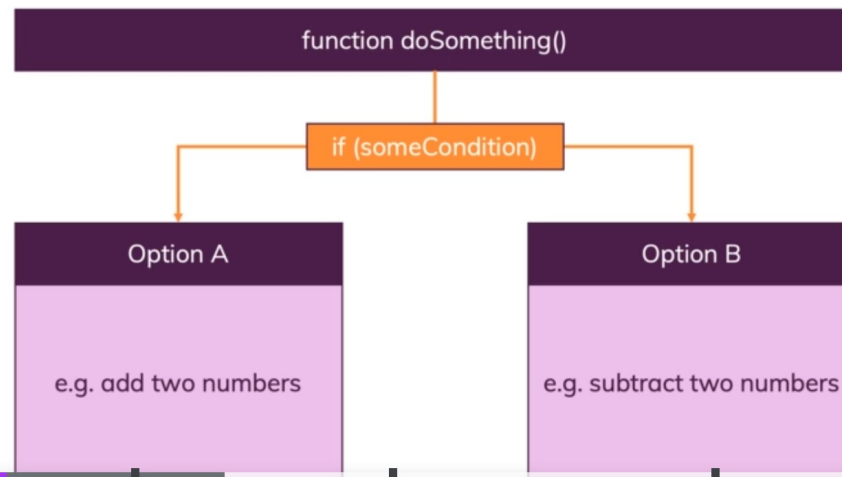


Falsy and Truthy Values

JavaScript Conditions work with Booleans (true / false) OR with "falsy" / "truthy" values



Conditional Code Execution



Falsy and Truthy Values

0	→	false
ANY other number (incl. negative numbers)	→	true
"" (empty string)	→	false
ANY other non-empty string (incl. "false")	→	true
{}, [] & all other objects or arrays	→	true
null, undefined, NaN	→	false

Loops

Execute code multiple times

for loop

Execute code a certain amount of times (with counter variable)

```
for (let i = 0; i < 3; i++)  
{  
  console.log(i);  
}
```

for-of loop

Execute for every element in an array

```
for (const el of array)  
{  
  console.log(el);  
}
```

for-in loop

Execute for every key in an object

```
for (const key in obj) {  
  console.log(key);  
  console.log(obj[key]);  
}
```

while loop

Execute code as long as a condition is true

```
while (isLoggedIn) {  
  ...  
}
```


Conditional Expressions / Ternary Operator

if statements return no values!

This will NOT work!

```
const userName = if (isLogin) {  
  return 'Max';  
} else {  
  return null;  
}
```

Use the ternary operator in such cases

```
const userName = isLogin ? 'Max' : null
```

Condition (can be written
exactly like in if statements)

Value if condition is
true / truthy

Value if condition is
false / falsy

Error Handling

Some errors can't be avoided (beyond your control as a developer)

User Input Errors

e.g. user enters text like 'hi' instead of a number

Network Errors

e.g. server is offline

...

Throw and catch errors to fail gracefully or recover if possible

```
try { ... } catch (error) { ... }
```